

## REPLICATOR DYNAMICS IN COMBINATORIAL OPTIMIZATION

Replicator equations are a class of dynamical systems developed and studied in the context of evolutionary game theory, a discipline pioneered by J. Maynard Smith [36] which aims to model the evolution of animal behavior using the principles and tools of game theory. Because of their dynamical properties, they have been recently applied with significant success to a number of combinatorial optimization problems. It is the purpose of this article to provide a summary and an up-to-date bibliography of these applications.

**The model and its properties.** In this section we discuss the basic intuition behind replicator equations and present a few theoretical properties that are instrumental for their application to optimization problems. For a more systematic treatment see [23, 55].

Consider a large, ideally infinite population of individuals belonging to the same species which compete for a particular limited resource, such as food, territory, etc. This kind of conflict is modeled as a game, the players being pairs of randomly selected population members. In contrast to traditional application fields of game theory, such as economics or sociology [33], players here do not behave “rationally,” but act instead according to a pre-programmed behavior pattern, or *pure strategy*. Reproduction is assumed to be asexual, which means that, apart from mutation, offspring will inherit the same genetic material, and hence behavioral phenotype, as its parent. Let  $J = \{1, \dots, n\}$  be the set of pure strategies and, for all  $i \in J$ , let  $x_i(t)$  be the relative frequency of population members playing strategy  $i$ , at time  $t$ . The *state* of the system at time  $t$  is simply the vector  $\mathbf{x}(t) = (x_1(t), \dots, x_n(t))^T$ . Clearly, the states are constrained to lie in the standard simplex of the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ :

$$S_n = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0 \forall i \in J, \mathbf{e}^T \mathbf{x} = 1\}.$$

Here and in the sequel, the letter  $\mathbf{e}$  is reserved for a vector of appropriate length, consisting of unit entries (hence  $\mathbf{e}^T \mathbf{x} = \sum_i x_i$ ).

One advantage of applying game theory to biology is that the notion of “utility” is much simpler and clearer than in human contexts. Here, a player’s utility can simply be measured in terms of Darwinian fitness or reproductive success, i.e., the player’s expected number of offspring. Let  $W = (w_{ij})$  be the  $n \times n$  “payoff” (or fitness) matrix. Specifically, for each pair of strategies  $i, j \in J$ ,  $w_{ij}$  represents the payoff of an individual playing strategy  $i$  against an opponent playing strategy  $j$ . Without loss of generality, we shall assume that the payoff matrix is non-negative, i.e.,  $w_{ij} \geq 0$  for all  $i, j \in J$ . At time  $t$ , the average payoff of strategy  $i$  is given by:

$$\pi_i(t) = \sum_{j=1}^n w_{ij} x_j(t) \quad (1)$$

while the mean payoff over the entire population is  $\sum_{i=1}^n x_i(t) \pi_i(t)$ .

In evolutionary game theory the assumption is made that the game is played over and over, generation after generation, and that the action of natural selection will result in the evolution of the fittest strategies. If successive generations blend into each other, the evolution of behavioral phenotypes can be described by the following set of differential equations [53]:

$$\dot{x}_i(t) = x_i(t) \left( \pi_i(t) - \sum_{j=1}^n x_j(t) \pi_j(t) \right) \quad (2)$$

for  $i = 1 \dots n$ , where a dot signifies derivative with respect to time. The basic idea behind this model is that the average rate of increase  $\dot{x}_i(t)/x_i(t)$  equals the difference between the average fitness of strategy  $i$  and the mean fitness over the entire population. It is straightforward to show that the simplex  $S_n$  is invariant under equation (2) or, in other words, any trajectory starting in  $S_n$  will remain in  $S_n$ . To see this, simply note that  $\frac{d}{dt} \sum_i x_i(t) = \sum_i \dot{x}_i(t) = 0$ , which means that the (relative) interior of  $S_n$  (i.e., the set defined by  $x_i > 0$ , for all  $i = 1 \dots n$ ) is invariant. The additional observation that the boundary too is invariant, completes the proof.

Similar arguments provide a rationale for the following discrete-time version of the replicator dynamics, assuming non-overlapping generations, which can be obtained from (2) by setting

$$1/\Delta t = \sum_{j=1}^n x_j(t)\pi_j(t):$$

$$x_i(t + \Delta t) = \frac{x_i(t)\pi_i(t)}{\sum_{j=1}^n x_j(t)\pi_j(t)} \quad (3)$$

for  $i = 1 \dots n$ . Because of the non-negativity of the fitness matrix  $W$  and the normalization factor, this system too makes the simplex  $S_n$  invariant as its continuous counterpart.

A point  $\mathbf{x} = \mathbf{x}(t)$  is said to be a *stationary* (or equilibrium) point for our dynamical systems, if  $\dot{x}_i(t) = 0$  in the continuous-time case, and  $x_i(t + \Delta t) = x_i(t)$  in the discrete-time case ( $i = 1 \dots n$ ). Moreover, a stationary point is said to be *asymptotically stable* if any trajectory starting in its vicinity will converge to it as  $t \rightarrow \infty$ . It turns out that both the continuous-time and discrete-time replicator dynamics have the same set of stationary points, namely all the points in  $S_n$  satisfying, for all  $i = 1 \dots n$ , the condition

$$x_i(t) \left( \pi_i(t) - \sum_{j=1}^n x_j(t)\pi_j(t) \right) = 0$$

or, equivalently,  $\pi_i(t) = \sum_{j=1}^n x_j(t)\pi_j(t)$  whenever  $x_i > 0$ .

Equations (2) and (3) arise independently in different branches of theoretical biology [23]. In population ecology, for example, the famous Lotka-Volterra equations for predator-prey systems turn out to be equivalent to the continuous-time dynamics (2), under a simple barycentric transformation and a change in velocity. In population genetics they are known as *selection equations* [17]. In this case, each  $x_i$  represents the frequency of the  $i$ -th allele  $A_i$  and the payoff  $w_{ij}$  is the fitness of genotype  $A_i A_j$ . Here the fitness matrix  $W$  is always symmetric. The discrete-time dynamical equations turn out to be a special case of a general class of dynamical systems introduced by Baum and Eagon [2] and studied by Baum and Sell [3] in the context of Markov chain theory. They also represent an instance of the so-called *relaxation labeling processes*, a class of parallel, distributed algorithms developed in computer vision to solve (continuous) constraint satisfaction problems [50, 25, 44]. An independent connection between relaxation labeling processes

and game theory has recently been described by Miller and Zucker [37].

The following theorem states that under replicator dynamics the population's average fitness always increases, provided that the payoff matrix is symmetric (in game theory terminology, this situation is referred to as a doubly symmetric game).

**Theorem 1** *Suppose that the (nonnegative) payoff matrix  $W$  is symmetric. Then, the quadratic polynomial  $F$  defined as*

$$F(\mathbf{x}) = \mathbf{x}^T W \mathbf{x} \quad (4)$$

*is strictly increasing along any non-constant trajectory of both continuous-time (2) and discrete-time (3) replicator equations. In other words, for all  $t \geq 0$  we have  $\frac{d}{dt}F(\mathbf{x}(t)) > 0$  for system (2), and  $F(\mathbf{x}(t + \Delta t)) > F(\mathbf{x}(t))$  for system (3), unless  $\mathbf{x}(t)$  is a stationary point. Furthermore, any such trajectory converges to a (unique) stationary point.*

The previous result is known in mathematical biology as the fundamental theorem of natural selection [17, 23, 55] and, in its original form, traces back to Fisher [18]. As far as the discrete-time model is concerned, it can be regarded as a straightforward implication of the Baum-Eagon theorem [2, 3] which is valid for general polynomial functions over product of simplices. Waugh and Westervelt [54] also proved a similar result for a related class of continuous- and discrete-time dynamical systems. In the discrete-time case, however, they put bounds on the eigenvalues of  $W$  in order to achieve convergence to fixed points.

The fact that all trajectories of the replicator dynamics converge to a stationary point has been proved more recently [32, 34]. However, in general, not all stationary points are local maximizers of  $F$  on  $S_n$ . The vertices of  $S_n$ , for example, are all stationary points for (2) and (3) whatever the landscape of  $F$ . Moreover, there may exist trajectories which, starting from the interior of  $S_n$ , eventually approach a saddle point of  $F$ . However, a result recently proved by Bomze [5] asserts that all asymptotically stable stationary points of replicator dynamics correspond to (strict) local maximizers of  $F$  on  $S_n$ ,

and vice versa (see [10] for additional results relating the fields of optimization theory, evolutionary game theory and the qualitative behavior of dynamical systems).

Under continuous-time replicator dynamics, the trajectories approach their limits most efficiently in the sense that (2) is a gradient system if one uses the (non-Euclidean) Shahshahani metric [23] which, for any point  $\mathbf{u} \in S_n$ , is defined as

$$d_{\mathbf{u}}(\mathbf{x}, \mathbf{y}) = \sum_{i:u_i>0} \frac{1}{u_i} x_i y_i .$$

This efficiency result is called Kimura's maximum principle.

**Maximum clique problems.** Let  $G = (V, E)$  be an undirected graph, where  $V = \{1, \dots, n\}$  is the set of vertices and  $E \subseteq V \times V$  is the set of edges. The *order* of  $G$  is the number of its vertices, and its *size* is the number of edges. Two vertices  $i, j \in V$  are said to be *adjacent* if  $(i, j) \in E$ . The *adjacency matrix* of  $G$  is the  $n \times n$  symmetric matrix  $A_G = (a_{ij})$  defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

A subset  $C$  of vertices in  $G$  is called a *clique* if all its vertices are mutually adjacent, i.e., for all  $i, j \in C$  we have  $(i, j) \in E$ . A clique is said to be *maximal* if it is not contained in any larger clique, and *maximum* if it is the largest clique in the graph. The *clique number*, denoted by  $\omega(G)$ , is defined as the cardinality of the maximum clique. The maximum clique problem is to find a clique whose cardinality equals the clique number.

The maximum clique problem is a well-known example of combinatorial optimization problem, not only because it was one of the first problems shown to be *NP*-complete [19], but also for its theoretical as well as practical implications. Due to the inherent computational complexity of the problem, exact algorithms are guaranteed to return a solution only in a time which increases exponentially with the number of vertices in the graph, and this makes them inapplicable even to moderately large problem instances. Moreover, a series of recent theoretical results show that

the problem is in fact difficult to solve even in terms of approximation. Because of these negative results, much effort has recently been directed towards devising efficient heuristics for finding large cliques, for which no formal guarantee of performance may be provided, but are anyway of interest in practical applications. We refer to [8] for a recent survey of results concerning algorithms, complexity and applications of this problem.

In 1965, Motzkin and Straus [38] established a remarkable connection between the maximum clique problem and a certain quadratic programming problem. Consider the following quadratic function, sometimes called the *Lagrangian* of  $G$ :

$$f_G(\mathbf{x}) = \mathbf{x}^T A_G \mathbf{x} \quad (5)$$

and let  $\mathbf{x}^*$  be a global maximizer of  $f_G$  on  $S_n$ ,  $n$  being the order of  $G$ . In [38] it is proved that the clique number of  $G$  is related to  $f_G(\mathbf{x}^*)$  by the following formula:

$$\omega(G) = \frac{1}{1 - f_G(\mathbf{x}^*)} . \quad (6)$$

Additionally, it is shown that a subset of vertices  $C$  is a maximum clique of  $G$  if and only if its *characteristic vector*  $\mathbf{x}^C$ , which is the vector of  $S_n$  defined as

$$x_i^C = \begin{cases} 1/|C|, & \text{if } i \in C \\ 0, & \text{otherwise} \end{cases}$$

is a global maximizer of  $f_G$  on  $S_n$ . In [21, 47], the Motzkin-Straus theorem has been extended by providing a characterization of *maximal* cliques in terms of *local* maximizers of  $f_G$  on  $S_n$ .

Once that the maximum clique problem is formulated in terms of maximizing a quadratic polynomial over the standard simplex, the use of replicator dynamics naturally suggests itself [42]. In fact, consider a replicator system with payoff matrix defined as:

$$W = A_G .$$

From the fundamental theorem of natural selection, we know that the replicator dynamical systems, starting from an arbitrary initial state, will iteratively maximize the Lagrangian  $f_G$  in  $S_n$ , and will eventually converge to a local maximizer which, by virtue of the Motzkin-Straus

formula provides an estimate of the clique number of  $G$ . Additionally, if the converged solution happens to be a characteristic vector of some subset of vertices of  $G$ , then we are also able to extract the vertices comprising the clique from its nonzero components. Clearly, in theory there is no formal guarantee that the converged solution will be a *global* maximizer of  $f_G$ . However, experimental work suggests that the basins of attraction of global maximizers are quite large, and frequently the algorithm converges to one of them.

In [42], Pelillo presents extensive experimental results with the previous approach over thousands of randomly generated graphs. The discrete-time dynamics (3) was used, and the system was started from the vector  $(1/n, \dots, 1/n)^T$  which corresponds to simplex barycenter. Two series of experiments were conducted. In the first one, graphs with a relatively small number of vertices were considered, i.e. with up to 500 vertices and densities ranging from 0.10 to 0.90. The solutions found by the algorithm were always very close to the optimal ones, as found by standard exact algorithms. In the second part of the study, graphs with up to 2000 vertices and about one million edges were used (in this case all graphs had density 0.50). Here to gauge the quality of the solutions found the Matula's estimate was employed, which accurately predicts the clique number in a random graph, when the number of vertices is sufficiently large [35]. Specifically, let

$$M(n, \delta) = 2 \log_{1/\delta} n - 2 \log_{1/\delta} \log_{1/\delta} n + 2 \log_{1/\delta} \frac{e}{2} + 1 .$$

Matula proved that, as  $n \rightarrow \infty$ , the order of the maximum clique in an  $n$ -vertex  $\delta$ -density random graph is either  $\lfloor M(n, \delta) \rfloor$  or  $\lceil M(n, \delta) \rceil$  with probability tending to 1, where  $\lfloor x \rfloor$  denotes the largest integer less than or equal to  $x$ , and  $\lceil x \rceil$  denotes the smallest integer greater than or equal to  $x$ . Interestingly, it was also shown that the smallest maximal clique is expected to have  $M(n, \delta)/2$  vertices [4]. Experimentally, in [42] it was found that the cardinality of the

cliques found by the replicator dynamical system turned out to be significantly larger than the estimated minimum, thereby contradicting what is known as the Jerrum conjecture [27], which states that in a large 0.5-density random graph it may be hard to find a clique whose order is even a bit larger than that of the smallest maximal clique. A similar conclusion was also drawn by Jagota [26]. Overall, the results presented in [42] were competitive with those obtained using more sophisticated neural network heuristics, both in terms of quality of solutions and speed.

One drawback associated with the original Motzkin-Straus formulation, however, relates to the existence of spurious solutions, i.e., maximizers of  $f_G$  which are not in the form of characteristic vectors. This was first observed by Pardalos and Phillips [40]. To illustrate, consider the path  $P^3$ , i.e. the graph with three vertices  $\{1, 2, 3\}$  and two edges, one between 1 and 2, and the other between 2 and 3. Clearly  $C = \{1, 2\}$  and  $D = \{2, 3\}$  are maximum cliques, and from the Motzkin-Straus theorem it follows that their characteristic vectors  $\mathbf{x}^C$  and  $\mathbf{x}^D$  are global maximizers of the Lagrangian of  $P^3$  in  $S_3$ . However, it can easily be proved that all the points lying on the segment connecting  $\mathbf{x}^C$  and  $\mathbf{x}^D$ , which is a subset of  $S_3$  since the simplex is convex, are also global solutions of the Motzkin-Straus program. Pelillo and Jagota [47] have recently provided general characterizations of such spurious solutions. In principle, spurious solutions represent a problem since, while providing information about the cardinality of the maximum clique, they do not allow us to easily extract its vertices.

The spurious solution problem has recently been solved by Bomze [5]. Consider the following regularized version of function  $f_G$ :

$$\hat{f}_G(\mathbf{x}) = \mathbf{x}^T A_G \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{x} \quad (7)$$

which is obtained from (5) by substituting the adjacency matrix  $A_G$  of  $G$  with

$$\hat{A}_G = A_G + \frac{1}{2} I_n$$

where  $I_n$  is the  $n \times n$  identity matrix. Unlike the Motzkin-Straus formulation, it can be proved

that *all* maximizers of  $\hat{f}_G$  on  $S_n$  are strict, and are characteristic vectors of maximal/maximum cliques in the graph [5].

**Theorem 2** *Let  $C$  be a subset of vertices of a graph  $G$ , and let  $\mathbf{x}^C$  be its characteristic vector. Then,  $C$  is a maximum (maximal) clique of  $G$  if and only if  $\mathbf{x}^C$  is a global (local) maximizer of  $\hat{f}_G$  in  $S_n$ . Moreover, all local (and hence global) maximizers of  $\hat{f}_G$  over  $S_n$  are strict.*

In an exact sense, therefore, a one-to-one correspondence exists between maximal cliques and local maximizers of  $\hat{f}_G$  in  $S_n$  on the one hand and maximum cliques and global maximizers on the other hand.

Preliminary experiments with this regularized formulation (7) on random graphs are reported in [5], and a more extensive empirical study on DIMACS benchmark graphs is presented in [10]. The emerging picture is the following. The solutions produced by the replicator models are typically very close to the ones obtained using more sophisticated continuous-based heuristics. Moreover, the original version of the Motzkin-Straus problem performs slightly better than its regularized counterpart, but the former often returns spurious solutions. This may be intuitively explained by observing that, since all local maxima of  $\hat{f}_G$  are strict, its landscape is certainly less smoothed than the one associated to the non-regularized version. This therefore enhances the tendency of local optimization procedures to get stuck into local maxima. This is the price to pay for the algorithm to return non-spurious, “informative” solutions.

In order to study the effects of varying the starting point of clique finding replicator dynamics, Bomze and Rendl [12] implemented various sophisticated heuristics and compared them with the usual (less expensive) strategy of starting from the simplex barycenter. Surprisingly, they concluded that the amount of sophistication seems to have no significant impact on the quality of the solutions obtained. Additionally, they showed that using (Runge-Kutta discretizations of) the continuous-time dynamics (2) instead of (3) does not improve efficiency.

This analysis indicates that to improve the performance of replicator dynamics on the maximum clique problem one has necessarily to resort to some escape strategies. Various attempts along this direction can be found in [5, 6, 9, 13].

Recently, the Motzkin-Straus theorem has been generalized to the weighted case [21]. Let  $G = (V, E, \mathbf{w})$  be a weighted graph, where  $V = \{1, \dots, n\}$  is the vertex set,  $E \subseteq V \times V$  is the edge set and  $\mathbf{w} \in \mathbb{R}^n$  is the *weight* vector, the  $i$ -th component of which corresponds to the weight assigned to vertex  $i$ . It is assumed that  $w_i > 0$  for all  $i \in V$ . Given a subset of vertices  $C$ , the weight assigned to  $C$  is defined as

$$W(C) = \sum_{i \in C} w_i .$$

A maximal weight clique  $C$  is one that is not contained in any other clique having weight larger than  $W(C)$ . Since we are assuming that all weights are positive, it is clear that the concepts of maximal clique and maximal weight clique coincide. A maximum weight clique is one having largest total weight, and the *weighted clique number* of  $G$ , denoted  $\omega(G, \mathbf{w})$ , is its weight. The maximum weight clique problem is to find a clique  $C$  such that  $W(C) = \omega(G, \mathbf{w})$  (see [8] for a recent review). The classical (unweighted) version of the maximum clique problem arises as a special case when all vertices have the same weight. For this reason the maximum weight clique problem has at least the same computational complexity as its unweighted counterpart.

Note that the original Motzkin-Straus program for unweighted graphs can be reformulated as a minimization problem by considering the function

$$g(\mathbf{x}) = \mathbf{x}^T (I + A_{\overline{G}}) \mathbf{x} \quad (8)$$

where  $A_{\overline{G}}$  is the adjacency matrix of the complement graph  $\overline{G}$ , which is the graph having the same vertex set as  $G$  and  $\overline{E} = \{(i, j) \in V \times V : i \neq j \text{ and } (i, j) \notin E\}$  as its edge set. It is straightforward to see that if  $\mathbf{x}^*$  is a global minimizer of  $g$  in  $S_n$ , then  $\omega(G) = 1/g(\mathbf{x}^*)$ . This is simply a different formulation of the Motzkin-Straus formula (6). Now, consider a weighted graph  $G = (V, E, \mathbf{w})$ , and let

$\mathcal{M}(G, \mathbf{w})$  be the class of symmetric  $n \times n$  matrices  $M = (m_{ij})_{i,j \in V}$  defined as  $2m_{ij} \geq m_{ii} + m_{jj}$  if  $(i, j) \notin E$  and  $m_{ij} = 0$  otherwise, and  $m_{ii} = \frac{1}{w_i}$  for all  $i \in V$ . Given a global solution  $\mathbf{x}^*$  of the following quadratic program, which is in general indefinite,

$$\begin{aligned} & \text{minimize} && g(\mathbf{x}) = \mathbf{x}^T M \mathbf{x} \\ & \text{subject to} && \mathbf{x} \in S_n \end{aligned} \quad (9)$$

we have [21]:

$$\omega(G, \mathbf{w}) = \frac{1}{g(\mathbf{x}^*)} \quad (10)$$

for any matrix  $M \in \mathcal{M}(G, \mathbf{w})$ . Furthermore, denote by  $\mathbf{x}^C(\mathbf{w})$  the *weighted characteristic vector* of  $C$ , which is the vector in  $S_n$  with coordinates

$$x_i^C(\mathbf{w}) = \begin{cases} w_i/W(C) & \text{if } i \in C, \\ 0 & \text{otherwise.} \end{cases}$$

It turns out that a subset  $C$  of vertices is a maximum weight clique if and only if its characteristic vector  $\mathbf{x}^C(\mathbf{w})$  is a global minimizer of (9). Notice that the matrix  $I + A_{\overline{G}}$  belongs to  $\mathcal{M}(G, \mathbf{e})$ . In other words, the original Motzkin-Straus theorem turns out to be a special case of the preceding result.

As in the unweighted case, this formulation suffers from the existence of spurious solutions, and this entails the lack of a one-to-one correspondence between the solutions of the continuous optimization problem and those of the original, discrete one. In [11] these spurious solutions are characterized and a regularized version which avoids this kind of problems is introduced (see also [7]). Specifically, let  $\mathcal{N}(G, \mathbf{w})$  be the the class of  $n \times n$  symmetric matrices  $M = (m_{ij})_{i,j \in V}$  defined as  $m_{ij} \geq m_{ii} + m_{jj}$  if  $(i, j) \notin E$  and  $m_{ij} = 0$  otherwise, and  $m_{ii} = \frac{1}{2w_i}$  for all  $i \in V$ . The following theorem is the weighted counterpart of Theorem 2.

**Theorem 3** *Let  $C$  be a subset of vertices of a weighted graph  $G = (V, E, \mathbf{w})$ , and let  $\mathbf{x}^C(\mathbf{w})$  be its characteristic vector. Then, for any matrix  $M \in \mathcal{N}(G, \mathbf{w})$ ,  $C$  is a maximum (maximal) weight clique of  $G$  if and only if  $\mathbf{x}^C(\mathbf{w})$  is a global (local) solution of program (9). Moreover, all local (and hence global) solutions of (9) are strict.*

Note that  $\mathcal{N}(G, \mathbf{w})$  is isomorphic to the positive orthant in  $\binom{n}{2} - |E|$  dimensions. This class is

a polyhedral pointed cone with its apex given by the matrix  $M(\mathbf{w}) = (m_{ij}(\mathbf{w}))_{i,j \in V}$  with entries

$$m_{ij}(\mathbf{w}) = \begin{cases} \frac{1}{2w_i} & \text{if } i = j, \\ \frac{1}{2w_i} + \frac{1}{2w_j} & \text{if } i \neq j, (i, j) \notin E, \\ 0 & \text{if } i \neq j, (i, j) \in E. \end{cases}$$

Observe that in the unweighted case,  $M(\mathbf{e}) = \mathbf{e}\mathbf{e}^T - \hat{A}_G = \hat{A}_{\overline{G}}$ , the regularized adjacency matrix of the complement graph  $\overline{G}$ . This reflects the elementary property that an independent set of  $G$ , i.e. a subset of pairwise non-adjacent vertices, is a clique of  $\overline{G}$ . Hence, while the local maximizers of  $\mathbf{x}^T \hat{A}_G \mathbf{x}$  over  $S_n$  correspond to maximal cliques of  $G$ , the local minimizers of  $\mathbf{x}^T \hat{A}_G \mathbf{x}$  over  $S_n$  correspond to maximal independent sets.

Theorem 3 suggests using replicator equations to approximately solve the maximum weight clique problem. Indeed, note that replicator equations are maximization procedures, while ours is a minimization problem. However, it is a straightforward exercise to see that the problem of minimizing a quadratic form  $\mathbf{x}^T M \mathbf{x}$  on  $S_n$  is equivalent to maximizing  $\gamma \mathbf{e}\mathbf{e}^T - M$ , where  $\gamma$  is an arbitrary constant. Therefore, the payoff matrix for replicator dynamics to be used in this case is:

$$W = \gamma \mathbf{e}\mathbf{e}^T - M$$

where  $M = (m_{ij})$  is any matrix in  $\mathcal{N}(G, \mathbf{w})$ , and

$$\gamma = \max_{i,j \in V} m_{ij}.$$

Experiments with this approach on both random graphs and DIMACS benchmark graphs are reported in [11]. Weights were generated randomly in both cases. The results obtained with replicator dynamics (3) were compared with those produced by a very efficient maximum weight clique algorithm of the branch-and-bound variety. The algorithm performed remarkably well especially on large and dense graphs, and it was typically an order of magnitude more efficient than its competitor.

**Graph isomorphism.** Given two graphs  $G' = (V', E')$  and  $G'' = (V'', E'')$ , an *isomorphism* between them is any bijection  $\phi : V' \rightarrow V''$  such that  $(i, j) \in E' \Leftrightarrow (\phi(i), \phi(j)) \in E''$ , for all  $i, j \in V'$ . Two graphs are said to be *isomorphic* if there exists an isomorphism between them.

The graph isomorphism problem is therefore to decide whether two graphs are isomorphic and, in the affirmative, to find an isomorphism.

The graph isomorphism problem is one of those few combinatorial optimization problems which still resist any computational complexity characterization [19, 28]. Despite decades of active research, no polynomial-time algorithm for it has yet been found. At the same time, while clearly belonging to  $NP$ , no proof has been provided that it is  $NP$ -complete. Indeed, there is strong evidence that this cannot be the case, for otherwise the polynomial hierarchy would collapse [14, 52]. The current belief is that the problem lies strictly between the  $P$  and  $NP$ -complete classes.

The subgraph isomorphism problem is more general and in fact more difficult, being  $NP$ -complete [19]. Given two graphs, it is the problem of determining whether one is isomorphic to a subgraph of the other. At the highest level of generality we find the maximum common subgraph problem, which consists of finding the largest isomorphic subgraphs of two graphs. A simpler version of this problem is to find a maximal common subgraph, i.e., an isomorphism between subgraphs which is not included in any larger subgraph isomorphism.

Barrow and Burstall [1], and also Kozen [30], introduced the notion of an *association graph* as a useful auxiliary graph structure for solving general graph/subgraph isomorphism problems. Specifically, the association graph derived from graphs  $G' = (V', E')$  and  $G'' = (V'', E'')$  is the undirected graph  $G = (V, E)$  where

$$V = V' \times V''$$

and

$$E = \{((i, h), (j, k)) \in V \times V : i \neq j, h \neq k \text{ and } (i, j) \in E' \Leftrightarrow (h, k) \in E''\}.$$

The following straightforward result establishes an equivalence between the graph isomorphism problem and the maximum clique problem [46].

**Theorem 4** *Let  $G' = (V', E')$  and  $G'' = (V'', E'')$  be two graphs of order  $n$ , and let  $G$  be the corresponding association graph. Then,  $G'$*

*and  $G''$  are isomorphic if and only if  $\omega(G) = n$ . In this case, any maximum clique of  $G$  induces an isomorphism between  $G'$  and  $G''$ , and vice versa. In general, maximum (maximal) cliques in  $G$  are in one-to-one correspondence with maximum (maximal) common subgraph isomorphisms between  $G'$  and  $G''$ .*

By virtue of Theorem 2, it is a straightforward exercise to formulate the graph isomorphism problem in terms of a quadratic programming problem. Let  $G'$  and  $G''$  be two arbitrary graphs of order  $n$ , and let  $A_G$  denote the adjacency matrix of the corresponding association graph  $G$ , whose order is  $n^2$ . The graph isomorphism problem is equivalent to the following program:

$$\begin{aligned} & \text{maximize} && \hat{f}_G(\mathbf{x}) = \mathbf{x}^T(A + \frac{1}{2}I_{n^2})\mathbf{x} \\ & \text{subject to} && \mathbf{x} \in S_{n^2} \end{aligned} \quad (11)$$

More precisely,  $G'$  and  $G''$  are isomorphic if and only if  $\hat{f}_G(\mathbf{x}^*) = 1 - 1/2n$ . In this case, any global solution to (11) induces an isomorphism between  $G'$  and  $G''$ , and vice versa. In general, local (global) solutions to (11) are in one-to-one correspondence with maximal (maximum) common subgraph isomorphisms between  $G'$  and  $G''$ .

The previous result allows one to use replicator dynamics with payoff matrix

$$W = A_G + \frac{1}{2}I_{n^2}$$

as a heuristic for graph isomorphism problems. Starting from an arbitrary initial state, the dynamical system will converge to a local solution of (11). This will correspond to a characteristic vector of a maximal clique in the association graph  $G$  which, in turn, will induce an isomorphism between two subgraphs of  $G'$  and  $G''$  which is maximal, in the sense that there is no other isomorphism between subgraphs of  $G'$  and  $G''$  that includes the one found.

The algorithm outlined above has been tested over hundreds of random 100-vertex graphs with expected densities ranging from 1% to 99%. Except for very sparse and very dense instances, the algorithm was always able to obtain a correct isomorphism very efficiently. In terms of

quality of solutions, the result compare favorably with those obtained using more sophisticated state-of-the-art deterministic annealing heuristics which, in contrast to replicator dynamics, are explicitly designed to escape from poor local solutions. As far as computational time is concerned, replicator dynamics turned out to be significantly faster.

In [46] experiments were also done using the following exponential version of replicator equations, which arises as a model of evolution guided by imitation [22, 23, 24, 55]:

$$\dot{x}_i(t) = x_i(t) \left( \frac{e^{\kappa\pi_i(t)}}{\sum_{j=1}^n x_j(t)e^{\kappa\pi_j(t)}} - 1 \right), \quad (12)$$

$i = 1 \dots n$ , where  $\kappa$  is a positive constant. As  $\kappa$  tends to 0, the orbits of this dynamics approach those of the standard, “first-order” replicator model (2), slowed down by the factor  $\kappa$ ; moreover, for large values of  $\kappa$  the model approximates the so-called “best-reply” dynamics [24]. As it turns out [22], these models behave essentially in the same way as the standard replicator equations (2), the only difference being the size of the basins of attraction around stable equilibria.

A customary way of discretizing equation (12) is given by the following difference equations [15, 20]:

$$x_i(t+1) = \frac{x_i(t)e^{\kappa\pi_i(t)}}{\sum_{j=1}^n x_j(t)e^{\kappa\pi_j(t)}}, \quad (13)$$

$i = 1 \dots n$ . The extensive results reported in [46] with this dynamics show that exponential replicator dynamics may be considerably faster and even more accurate than the standard, first-order model.

The approach just described is general and can clearly be extended to deal with subgraph isomorphism or relational structure matching problems [45]. Preliminary experiments, however, seem to indicate that local optima may represent a problem here, especially in matching sparse and dense graphs. In these cases escape procedures like those presented in [5, 6, 9, 13] would be helpful.

**Subtree isomorphism.** Given a graph  $G = (V, E)$ , a *path* is any sequence of distinct vertices  $i_0 i_1 \dots i_n$  such that for all  $k = 1 \dots n$ ,  $(i_{k-1}, i_k) \in E$ ; in this case, the *length* of the path is  $n$ . If  $i_0 = i_n$  the path is called a *cycle*. A graph is said to be *connected* if any pair of vertices is joined by a path. The *distance* between two vertices  $i$  and  $j$ , denoted by  $d(i, j)$ , is the length of the shortest path joining them (by convention  $d(i, j) = \infty$ , if there is no such path). Given a subset of vertices  $C \subseteq V$ , the *induced subgraph*  $G[C]$  is the graph having  $C$  as its vertex set, and two vertices are adjacent in  $G[C]$  if and only if they are adjacent in  $G$ .

A connected graph with no cycles is called a *tree*. A *rooted tree* is one which has a distinguished vertex, called the *root*. The *level* of a vertex  $i$  in a rooted tree, denoted by  $\text{lev}(i)$ , is the length of the path connecting the root to  $i$ . Note that there is an obvious equivalence between rooted trees and directed trees, where the edges are assumed to be oriented. We shall therefore use the same terminology typically used for directed trees to define the relation between two adjacent vertices. In particular, if  $(i, j) \in E$  and  $\text{lev}(j) - \text{lev}(i) = +1$ , we say that  $i$  is the *parent* of  $j$  and, conversely,  $j$  is a *child* of  $i$ . Trees have a number of interesting properties. One which turns out to be very useful for our characterization is that in a tree any two vertices are connected by a *unique* path.

Let  $T_1 = (V_1, E_1)$  and  $T_2 = (V_2, E_2)$  be two rooted trees. Any bijection  $\phi : H_1 \rightarrow H_2$ , with  $H_1 \subseteq V_1$  and  $H_2 \subseteq V_2$ , is called a *subtree isomorphism* if it preserves the adjacency and hierarchical relationships between the vertices and, in addition, the subgraphs obtained when we restrict ourselves to  $H_1$  and  $H_2$ , i.e.,  $T_1[H_1]$  and  $T_2[H_2]$ , are trees. The former condition amounts to stating that, given  $i, j \in H_1$ , we have  $(i, j) \in E_1$  if and only if  $(\phi(i), \phi(j)) \in E_2$ , and  $i$  is the parent of  $j$  if and only if  $\phi(i)$  is the parent of  $\phi(j)$ . A subtree isomorphism is *maximal* if there is no other subtree isomorphism  $\phi' : H'_1 \rightarrow H'_2$  with  $H_1$  a strict subset of  $H'_1$ , and *maximum* if  $H_1$  has largest cardinality. The maximal (maximum) subtree isomorphism problem is to find



a maximal (maximum) subtree isomorphism between two rooted trees. This is a problem solvable in polynomial time [19].

Let  $i$  and  $j$  be two distinct vertices of a rooted tree  $T$ , and let  $i = x_0x_1 \dots x_n = j$  be the (unique) path joining them. The *path-string* of  $i$  and  $j$ , denoted by  $\text{str}(i, j)$ , is the string  $s_1s_2 \dots s_n$  on the alphabet  $\{-1, +1\}$  where, for all  $k = 1 \dots n$ ,  $s_k = \text{lev}(x_k) - \text{lev}(x_{k-1})$ . By convention, when  $i = j$  we define  $\text{str}(i, j) = \varepsilon$ , where  $\varepsilon$  is the null string (i.e., the string having zero length). The path-string concept has a very intuitive meaning. Because of the orientation induced by the root, only two types of elementary moves can be done from any given vertex, i.e., going down to one of the children (if one exists) or going up to the parent (if the vertex is not the root). Assigning to the first move the label  $+1$ , and to the second the label  $-1$ , the path-string of  $i$  and  $j$  is simply the string of elementary moves required to move from  $i$  to  $j$ , following the unique path joining them.

The *tree association graph* (TAG) of two rooted trees  $T_1 = (V_1, E_1)$  and  $T_2 = (V_2, E_2)$  is the graph  $G = (V, E)$  where

$$V = V_1 \times V_2 \quad (14)$$

and, for any two vertices  $(i, h)$  and  $(j, k)$  in  $V$ , we have

$$((i, h), (j, k)) \in E \Leftrightarrow \text{str}(i, j) = \text{str}(h, k) . \quad (15)$$

The following theorem establishes a one-to-one correspondence between the maximum subtree isomorphism problem and the maximum clique problem [48].

**Theorem 5** *Any maximal (maximum) subtree isomorphism between two rooted trees induces a maximal (maximum) clique in the corresponding TAG, and vice versa.*

In many practical applications the trees being matched have vertices with an associated vector of symbolic and/or numeric attributes. The framework just described can naturally be extended for solving attributed tree matching problems [48].

Formally, an *attributed tree* is a triple  $T = (V, E, \alpha)$ , where  $(V, E)$  is the “underlying” rooted tree and  $\alpha$  is a function which assigns an

attribute vector  $\alpha(i)$  to each vertex  $i \in V$ . It is clear that in matching two attributed trees, the objective is to find an isomorphism which pairs vertices having “similar” attributes. To this end, let  $\sigma$  be any similarity measure on the attribute space, i.e., any (symmetric) function which assigns a positive number to any pair of attribute vectors. If  $\phi : H_1 \rightarrow H_2$  is a subtree isomorphism between two attributed trees  $T_1 = (V_1, E_1, \alpha_1)$  and  $T_2 = (V_2, E_2, \alpha_2)$ , the overall similarity between the induced subtrees  $T_1[H_1]$  and  $T_2[H_2]$  can be defined as follows:

$$S(\phi) = \sum_{i \in H_1} \sigma(\alpha_1(i), \alpha_2(\phi(i))) .$$

The isomorphism  $\phi$  is called a *maximal similarity subtree isomorphism* if there is no other subtree isomorphism  $\phi' : H'_1 \rightarrow H'_2$  such that  $H_1$  is a strict subset of  $H'_1$  and  $S(\phi) < S(\phi')$ . It is called a *maximum similarity subtree isomorphism* if  $S(\phi)$  is largest among all subtree isomorphisms between  $T_1$  and  $T_2$ .

The *weighted TAG* of two attributed trees  $T_1$  and  $T_2$  is the weighted graph  $G = (V, E, \mathbf{w})$  where  $V$  and  $E$  are defined as in (14) and (15), and  $\mathbf{w}$  is a vector which assigns a positive weight to each vertex  $(i, h) \in V = V_1 \times V_2$  as follows:

$$w_{ih} = \sigma(\alpha_1(i), \alpha_2(h)) .$$

The following result is the weighted counterpart of Theorem 5 [48].

**Theorem 6** *Any maximal (maximum) similarity subtree isomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted TAG, and vice versa.*

Theorems 5 and 6 provide a formal justification for applying replicator dynamics to find maximal subtree isomorphisms. In [48] this approach has been applied in computer vision to the problem of matching articulated and deformed visual shapes described by “shock” trees, an abstract representation of shape based on the singularities arising during a curve evolution process. The experiments, conducted on a number of shapes representing various object classes, yielded very good results, both in the weighted

and in the unweighted case. The system typically converged towards the globally optimal solutions in only a few seconds, and compared favorably with another powerful tree matching algorithm.

**A geometric problem.** Let  $G = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$  be a finite set of points in  $\mathbb{R}^n$ . The convex hull of  $G$ , denoted by  $\text{conv}(G)$ , is defined as the smallest convex set containing  $G$ . A basic problem in computational geometry is to determine whether a given query point  $\mathbf{y}$  is inside or outside  $\text{conv}(G)$  [49]. This task can easily be accomplished by a replicator dynamical system [43]. Such an algorithm can be used as a sub-routine for solving more general geometric problems, such as the polygon inclusion and the convex hull problems.

Consider the  $n \times m$  real matrix defined as  $X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_m]$ . It is well known that  $\text{conv}(G)$  can be written as

$$\text{conv}(G) = \{\mathbf{u} \in \mathbb{R}^n : \mathbf{u} = X\mathbf{v}, \mathbf{v} \in S_m\} .$$

Given an arbitrary point  $\mathbf{y} \in \mathbb{R}^n$  the following measure

$$E(\mathbf{y}, G) = \min_{\mathbf{v} \in S_m} \|X\mathbf{v} - \mathbf{y}\|_2 , \quad (16)$$

sometimes called the *exteriority* of  $\mathbf{y}$  to  $\text{conv}(G)$ , is just the Euclidean distance between  $\mathbf{y}$  and its closest point in  $\text{conv}(G)$ . The exteriority measure can provide useful information about the ability of neural networks to generalize well [16]. Clearly,  $\mathbf{y} \in \text{conv}(G)$  if and only if  $E(\mathbf{y}, G) = 0$ , in which case the closest point to  $\mathbf{y}$  is  $\mathbf{y}$  itself.

For convenience, the problem of evaluating  $E(\mathbf{y}, G)$  is translated into the equivalent (but more manageable) quadratic program:

$$\begin{aligned} & \text{minimize} && C(\mathbf{v}) = \frac{1}{2} \|X\mathbf{v} - \mathbf{y}\|_2^2 \\ & \text{subject to} && \mathbf{v} \in S_m . \end{aligned} \quad (17)$$

It is a well-known fact that  $C$  is convex (strictly convex indeed if the vectors  $\mathbf{x}_1, \dots, \mathbf{x}_m$  happen to be linearly independent), and this implies that all local minima of  $C$  are also global minima. Any descent procedure is therefore guaranteed to approach the global optimal solution in this case, without the risk of getting trapped into poor local minima.

It is interesting to note that a similar optimization problem, known as the problem of “optimal stability,” also arises in the context of learning in perceptron neural networks, where the goal is to derive the network’s parameters so as to ensure larger basins of attraction [31, 51]. Moreover, our problem turns out to be closely related to that of determining whether a given set of prototype vectors can be stored in a neural network associative memory [29].

Note that the quadratic objective function in (17) is explicitly written as follows:

$$C(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T X^T X \mathbf{v} - \mathbf{y}^T X \mathbf{v} + \frac{1}{2} \mathbf{y}^T \mathbf{y}$$

which is a non-homogeneous quadratic polynomial. In order for replicator equations to find a solution of problem (17), we need to construct the payoff matrix as

$$W = X^T X$$

and to replace the  $\pi$  function defined in (1) with:

$$\pi_i(t) = \sum_{j=1}^n w_{ij} x_j(t) + s_i$$

where  $s_i$  equals the  $i$ -th component of  $-X^T \mathbf{y}$ . After a proper rescaling of  $W$  and the  $s_i$ ’s, it is readily seen that  $C$  is a Liapunov function for both continuous-time (2) and discrete-time (3) dynamics. The algorithms will converge to a local solution of (17), say  $\mathbf{v}^*$ , starting from any interior point. Owing to the convexity of  $C$ ,  $\mathbf{v}^*$  will be also a global minimizer of  $C$ , so that the exteriority can be calculated as:

$$E(\mathbf{y}, G) = \sqrt{2C(\mathbf{v}^*)} .$$

In [43], experiments with a simple toy problem demonstrate the validity of the approach.

**Multi-population models.** The single-population replicator equations discussed so far can easily be generalized to the case where interactions take place among  $n \geq 2$  individuals randomly drawn from  $n$  distinct populations [23, 55]. In this case, the continuous-time dynamics (2) becomes

$$\dot{x}_i^\mu(t) = x_i^\mu(t) \left( \pi_i^\mu(t) - \sum_{\nu} x_i^\nu(t) \pi_i^\nu(t) \right), \quad (18)$$

and its discrete-time counterpart is

$$x_i^\mu(t + \Delta t) = \frac{x_i^\mu(t)\pi_i^\mu(t)}{\sum_\nu x_i^\nu(t)\pi_i^\nu(t)}. \quad (19)$$

The  $\pi$  function can either be linear, as in (1), or can take a more general form. If there exists a polynomial  $F$  such that

$$\pi_i^\mu = \frac{\partial F}{\partial x_i^\mu},$$

then it can be proved that  $F$  strictly increases along any trajectory of both dynamics [23, 2, 3]. Note that these dynamics work in a product of standard simplices.

Mühlenbein *et al.* [39] used multi-population replicator equations to approximately solve the graph partitioning problem, which is  $NP$ -complete [19]. Given a graph  $G = (V, E)$  with edge weights  $w_{ij}$ , their goal was to partition the vertices of  $G$  into a predefined number of clusters in such a way as to maximize the overall intra-partition traffic

$$F = \prod_{\mu} K^{\mu}, \quad (20)$$

where

$$K^{\mu} = \sum_i \sum_j w_{ij} x_i^{\mu} x_j^{\mu} \quad (21)$$

is the intra-partition traffic for cluster  $\mu$ . Here,  $x_i^{\mu}$  can be interpreted as the probability that vertex  $i$  belongs to cluster  $\mu$ .

By putting

$$\pi_i^{\mu} = \frac{2F \sum_j w_{ij} x_j^{\mu}}{K^{\mu}}$$

the replicator equations seen above will indeed converge toward a maximizer of  $F$ . However, in so doing the system typically converges towards an interior attractor, thereby giving an infeasible solution. To avoid this problem, Mühlenbein *et al.* [39] put a “selection pressure” parameter  $S$  on the main diagonal of the weight matrix, and altered it during the evolution of the process. Intuitively,  $S = 0$  has no influence on the system. Negative values of  $S$  prevent the vertices to decide for a partition, whereas positive values force the vertices to take a decision. The proposed algorithm starts with a negative value of  $S$ , and makes the discrete-time dynamics (19) evolve. After convergence, if an infeasible solution has

been found,  $S$  is increased and the algorithm is started again. The entire procedure is iterated until convergence to a feasible solution. A similar, but more principled, strategy for the maximum clique problem can be found in [9]. The results presented in [39] on a particular problem instance are fairly encouraging. However, more experiments on larger and diverse graphs are needed to fully assess the potential of the approach.

Multi-population replicator models have also been used in [39, 41] to solve the traveling salesman problem, which asks for the shortest closed tour connecting a given set of cities, subject to the constraint that each city be visited only once. The results presented on small problem instances, i.e., up to 30 cities, are encouraging but it seems that the results do not scale well with the size of the problem.

**Conclusions.** Despite their simplicity and inherent inability to escape from local solutions, replicator dynamics have proved to be a useful heuristic for attacking a variety of combinatorial optimization problems. They are completely devoid of operational parameters, which typically require a lengthy, problem-dependent tuning phase, and are especially suited for parallel hardware implementation.

## References

- [1] BARROW, H. G., AND BURSTALL, R. M.: ‘Subgraph isomorphism, matching relational structures and maximal cliques’, *Inform. Process. Lett.* **4**, no. 4 (1976), 83–84.
- [2] BAUM, L. E., AND EAGON, J. A.: ‘An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology’, *Bull. Amer. Math. Soc.* **73** (1967), 360–363.
- [3] BAUM, L. E., AND SELL, G. R.: ‘Growth transformations for functions on manifolds’, *Pacific J. Math.* **27**, no. 2 (1968), 211–227.
- [4] BOLLOBÁS, B., AND ERDŐS, P.: ‘Cliques in Random Graphs’, *Math. Proc. Cambridge Philos. Soc.* **80** (1976), 419–427.
- [5] BOMZE, I. M.: ‘Evolution towards the maximum clique’, *J. Glob. Optim.* **10** (1997), 143–164.
- [6] BOMZE, I. M.: ‘Global escape strategies for maximizing quadratic forms over a simplex’, *J. Glob. Optim.* **11** (1997), 325–338.
- [7] BOMZE, I. M.: ‘On standard quadratic optimization problems’, *J. Glob. Optim.* **13** (1998), 369–387.

- [8] BOMZE, I. M., BUDINICH, M., PARDALOS, P. M., AND PELILLO, M.: 'The maximum clique problem', *Handbook of Combinatorial Optimization (Supplement Volume A)*, in D.-Z. DU AND P. M. PARDALOS (eds.). Kluwer Academic Publishers, Boston, MA, 1999.
- [9] BOMZE, I. M., BUDINICH, M., PELILLO, M., AND ROSSI, C.: 'Annealed replication: A new heuristic for the maximum clique problem', *Discr. Appl. Math.* (2000), in press.
- [10] BOMZE, I. M., PELILLO, M., AND GIACOMINI, R.: 'Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale', in I. M. BOMZE, T. CSENDES, R. HORST, AND P. M. PARDALOS (eds.): *Developments in Global Optimization*, Kluwer Academic Publishers, 1997, pp. 95–108.
- [11] BOMZE, I. M., PELILLO, M., AND STIX, V.: Approximating the maximum weight clique using replicator dynamics, Tech. Rep. CS-99-13, Dipartimento di Informatica, Università Ca' Foscari di Venezia, 1999.
- [12] BOMZE, I. M., AND RENDL, F.: 'Replicator dynamics for evolution towards the maximum clique: Variations and experiments', in R. DE LEONE, A. MURLÍ, P. M. PARDALOS, AND G. TORALDO (eds.): *High Performance Algorithms and Software in Nonlinear Optimization*, Kluwer Academic Publishers, 1998, pp. 53–67.
- [13] BOMZE, I. M., AND STIX, V.: 'Genetic engineering via negative fitness: Evolutionary dynamics for global optimization', *Ann. Oper. Res.* **90** (1999), in press.
- [14] BOPPANA, R. B., HASTAD, J., AND ZACHOS, S.: 'Does co-NP have short interactive proofs?', *Inform. Process. Lett.* **25** (1987), 127–132.
- [15] CABRALES, A., AND SOBEL, J.: 'On the limit points of discrete selection dynamics', *J. Econom. Theory* **57** (1992), 407–419.
- [16] COURRIEU, P.: 'Three algorithms for estimating the domain of validity of feedforward neural networks', *Neural Networks* **7**, no. 1 (1994), 169–174.
- [17] CROW, J. F., AND KIMURA, M.: *An Introduction to Population Genetics Theory*, Harper & Row, New York, 1970.
- [18] FISHER, R. A.: *The Genetical Theory of Natural Selection*, Oxford University Press, London, UK, 1930.
- [19] GAREY, M. R., AND JOHNSON, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA, 1979.
- [20] GAUNERSDORFER, A., AND HOFBAUER, J.: 'Fictitious play, Shapley polygons, and the replicator equation', *Games Econom. Behav.* **11** (1995), 279–303.
- [21] GIBBONS, L. E., HEARN, D. W., PARDALOS, P. M., AND RAMANA, M. V.: 'Continuous characterizations of the maximum clique problem', *Math. Oper. Res.* **22** (1997), 754–768.
- [22] HOFBAUER, J.: Imitation dynamics for games, Collegium Budapest, preprint, 1995.
- [23] HOFBAUER, J., AND SIGMUND, K.: *Evolutionary Games and Population Dynamics*, Cambridge University Press, Cambridge, UK, 1998.
- [24] HOFBAUER, J., AND WEIBULL, J. W.: 'Evolutionary selection against dominated strategies', *J. Econom. Theory* **71** (1996), 558–573.
- [25] HUMMEL, R. A., AND ZUCKER, S. W.: 'On the foundations of relaxation labeling processes', *IEEE Trans. Pattern Anal. Machine Intell.* **5** (1983), 267–287.
- [26] JAGOTA, A.: 'Approximating maximum clique with a Hopfield neural network', *IEEE Trans. Neural Networks* **6** (1995), 724–735.
- [27] JERRUM, M.: 'Large cliques elude the Metropolis process', *Random Structures and Algorithms* **3** (1992), 347–359.
- [28] JOHNSON, D. S.: 'The NP-completeness column: An ongoing guide', *J. Algorithms* **9** (1988), 426–444.
- [29] KAMP, Y., AND HASLER, M.: *Recursive Neural Networks for Associative Memory*, John Wiley & Sons, New York, 1990.
- [30] KOZEN, D.: 'A clique problem equivalent to graph isomorphism', *SIGACT News* (1978), 50–52.
- [31] KRAUTH, W., AND MÉZARD, M.: 'Learning algorithms with optimal stability in neural networks', *J. Phys. A* **20** (1987), L745–L752.
- [32] LOSERT, V., AND AKIN, E.: 'Dynamics of games and genes: Discrete versus continuous time', *J. Math. Biol.* **17** (1983), 241–251.
- [33] LUCE, R. D., AND RAIFFA, H.: *Games and Decisions*, John Wiley & Sons, New York, 1957.
- [34] LYUBICH, YU., MAISTROWSKII, G. D., AND OL'KHOVSKII, YU. G.: 'Selection-induced convergence to equilibrium in a single-locus autosomal population', *Problems of Information Transmission* **16** (1980), 66–75.
- [35] MATULA, D. W.: The largest clique size in a random graph, Tech. Rep. CS 7608, Department of Computer Science, Southern Methodist University, 1976.
- [36] MAYNARD SMITH, J.: *Evolution and the Theory of Games*, Cambridge University Press, Cambridge, UK, 1982.
- [37] MILLER, D. A., AND ZUCKER, S. W.: 'Copositive-plus Lemke algorithm solves polymatrix games', *Oper. Res. Lett.* **10** (1991), 285–290.
- [38] MOTZKIN, T. S., AND STRAUS, E. G.: 'Maxima for graphs and a new proof of a theorem of Turán', *Canad. J. Math.* **17** (1965), 533–540.
- [39] MÜHLENBEIN, H., GORGES-SCHLEUTER, M., AND KRÄMER, O.: 'Evolution algorithms in combinatorial optimization', *Parallel Computing* **7** (1988), 65–85.

- [40] PARDALOS, P. M., AND PHILLIPS, A. T.: ‘A global optimization approach for solving the maximum clique problem’, *Int. J. Computer Math.* **33** (1990), 209–216.
- [41] PELILLO, M.: ‘Relaxation labeling processes for the traveling salesman problem’: *Proc. Int. J. Conf. Neural Networks*, 1993, pp. 2429–2432.
- [42] PELILLO, M.: ‘Relaxation labeling networks for the maximum clique problem’, *J. Artif. Neural Networks* **2** (1995), 313–328.
- [43] PELILLO, M.: ‘A relaxation algorithm for estimating the domain of validity of feedforward neural networks’, *Neural Process. Lett.* **3**, no. 3 (1996), 113–121.
- [44] PELILLO, M.: ‘The dynamics of nonlinear relaxation labeling processes’, *J. Math. Imaging Vision* **7**, no. 4 (1997), 309–323.
- [45] PELILLO, M.: ‘A unifying framework for relational structure matching’: *Proc. 14th Int. Conf. Pattern Recognition*, IEEE Computer Society Press, 1998, pp. 1316–1319.
- [46] PELILLO, M.: ‘Replicator equations, maximal cliques, and graph isomorphism’, *Neural Computation* **11**, no. 8 (1999), 2023–2045.
- [47] PELILLO, M., AND JAGOTA, A.: ‘Feasible and infeasible maxima in a quadratic program for maximum clique’, *J. Artif. Neural Networks* **2** (1995), 411–420.
- [48] PELILLO, M., SIDDIQI, K., AND ZUCKER, S. W.: ‘Matching hierarchical structures using association graphs’, *IEEE Trans. Pattern Anal. Machine Intell.* **21**, no. 11 (1999), 1105–1120.
- [49] PREPARATA, F. P., AND SHAMOS, M. I.: *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
- [50] ROSENFELD, A., HUMMEL, R. A., AND ZUCKER, S. W.: ‘Scene labeling by relaxation operations’, *IEEE Trans. Syst. Man & Cybern.* **6** (1976), 420–433.
- [51] RUJÁN, P.: ‘A fast method for calculating the perceptron with maximal stability’, *J. Phys. I France* **3** (1993), 277–290.
- [52] SCHÖNING, U.: ‘Graph isomorphism is in the low hierarchy’, *J. Comput. Syst. Sci.* **37** (1988), 312–323.
- [53] TAYLOR, P., AND JONKER, L.: ‘Evolutionarily stable strategies and game dynamics’, *Math. Biosci.* **40** (1978), 145–156.
- [54] WAUGH, F. R., AND WESTERVELT, R. M.: ‘Analog neural networks with local competition. I. Dynamics and stability’, *Phys. Rev. E* **47**, no. 6 (1993), 4524–4536.
- [55] WEIBULL, J. W.: *Evolutionary Game Theory*, MIT Press, Cambridge, MA, 1995.

*AMS 1991 Subject Classification:* 90C27, 90C20, 90C35, 90C59, 91A22, 37B25, 05C69, 05C60.

*Key words and phrases:* Quadratic programming, graphs, cliques, evolutionary game theory, heuristics, dynamical systems.

*Marcello Pelillo*

Università Ca’ Foscari di Venezia

Italy

*E-mail address:* pelillo@dsi.unive.it